

**REMARKS**

This Amendment, submitted in response to the Office Action dated September 29, 2006, is believed to be fully responsive to each point of rejection raised therein. Accordingly, favorable reconsideration on the merits is respectfully requested.

Claims 1-19 are now all of the claims pending in the application.

**I. Rejection of claims 8-10 under 35 U.S.C. § 101**

Claims 8-10 stand rejected under 35 U.S.C. § 101 because the claimed invention is allegedly directed to non-statutory subject matter. In particular, the Examiner asserts that a computer program product is non-statutory because software has to be embedded on a computer readable medium in order to be useful. The Examiner asserts that claims 8-10 lack any teaching about a computer program being stored in a memory. Applicant has amended claim 8 as indicated above. Applicant believes that the amendment to claim 8 addresses the 35 U.S.C. § 101 issue raised by the Examiner, consequently, Applicant requests that the 35 U.S.C. § 101 rejection of claims 8-10 be withdrawn.

**II. Rejection of claims 1, 3, 5, 7, 8 and 11 under 35 U.S.C. § 112**

Claims 1, 3, 5, 7, 8 and 11 stand rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. In particular, the Examiner asserts that the term “can” is not defined by the claim and the specification does not provide a standard for ascertaining the requisite degree. Applicant submits that the term “can” is not indefinite (see MPEP 2173.05); however, in order to expedite the prosecution for this application, Applicant has

amended claims 1, 3, 5, 7, 8, 10, 11 and 13 as indicated above. Consequently, Applicant requests that the rejection of claims under 35 U.S.C. § 112, second paragraph be withdrawn.

**III. Rejection of claims 1-10 and 16 under 35 U.S.C. § 103**

Claims 1-10 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi et al.<sup>1</sup> (U.S. Pub. No. 2005/0165820; hereinafter "Pudipeddi") in view of Goudie et al. (U.S. Pub. No. 2002/0156764; hereinafter "Goudie").

Claim 1 recites:

"grouping all of said data update requests into a plurality of blocks for execution by a data processor, the data update requests within each of said blocks and from one of said blocks to a next one of said blocks being arranged in an order that said data update requests need to be executed to yield a proper data result... and

then said data processor processing said data update requests within said one block in said order, and then said data processor processing said data update requests within said next block in said order."

The Examiner asserts that queues 402 of Pudipeddi teach the grouping of all data requests into a plurality of blocks. The queues 402 of Pudipeddi include items 406 which are requests to retrieve data stored in for example, an optical disk. However, the queues 402 (cited by the examiner for teaching the claimed blocks of data update requests) do not contain all of the data update requests. In particular, Pudipeddi is not concerned with grouping all requests for data. As illustrated in Figs. 8A-8E of Pudipeddi, requests for data stored in a storage media are

---

<sup>1</sup> Applicant notes that Pudipeddi '820 does not qualify as prior art under 35 U.S.C. § 102. However, Pudipeddi is a continuation of U.S. Patent No. 6,920,447 which qualifies as prior art under 35 U.S.C. § 102(e). Therefore, Applicant has traversed the rejection.

grouped into two monotonically increasing sequences. See para. 96. The reason Pudipeddi uses two monotonically increasing sequences is to address the situation where a request for a data object is dynamically queued during the reading of a medium. Therefore, in the event the location of a request has already passed its location on a first sequence, the request can then be inserted in a location in the second sequence. See para. 88. Thus, Pudipeddi discloses the insertion of individual requests which are not grouped in a block (queues 402 as cited by the Examiner). Since Pudipeddi is concerned with the dynamic insertion of requests, Pudipeddi does not teach grouping all of said data update requests into a plurality of blocks.

Claim 1 also recites “each of said blocks having approximately a **same capacity for said data update requests**, said **capacity corresponding to a number of said data update requests** which said data processor is adapted to efficiently process in order before processing said data update requests in the next one of said blocks.” The Examiner concedes that Pudipeddi does not teach this aspect of the claim cites Goudie to cure the deficiency.

In particular, the Examiner asserts that Goudie teaches a method of managing data in which the plurality of queues use a memory of the same size, citing para. 23 of Goudie in support. The aspect of Goudie cited by the Examiner discloses that all of the element queues in a grouped queue use memory blocks of the same size. However, although the blocks in a queue are the same size, there is no teaching or suggestion that the queues (blocks of update requests as asserted by the Examiner) of Goudie have the same capacity. In addition, although the memory blocks of Goudie are the same size, there is no teaching or suggestion that the capacity of the memory blocks correspond to a number of data update requests. Further, the size of the memory

blocks of Goudie have no relationship with a number of data update requests. In particular, there is no teaching or suggestion that the blocks of Goudie include data update requests.

The Examiner asserts it would have been obvious to use Pudipeddi's requests processing method and system for the update information because selecting particular type of requests to be processed by the processing unit is just an intended use and is not different from processing any other type of request. Further, the Examiner asserts that it would have been obvious to use the queues with the same capacity as taught by Goudie because predefining the queue's size is a design choice and that setting all the blocks to the same size is simpler than adjusting the individual size of the particular queue or block.

Contrary to the Examiner's assertions, the combination of Goudie with Pudipeddi is not obvious. Pudipeddi is not at all concerned with a data update. Pudipeddi is concerned with the recall of stored backup data. If Pudipeddi were modified as suggested by the Examiner to include data update requests, this would defeat the stored data retrieval of Pudipeddi. In particular, if update data requests were retrieved, at no point would stored backup data be recalled.

Further, Pudipeddi is directed to dynamic adjustment of queues. See Figs. 8A-8E. Therefore, Pudipeddi is directed to adjusting the size of a queue. Consequently, it would be contrary to the operation of Pudipeddi that the queue sizes be of a same capacity, further evidencing that the combination of Goudie with Pudipeddi is not obvious.

For at least the above reasons, claim 1 and its dependent claims should be deemed allowable. To the extent claims 5 and 8 recite similar subject claims, claims 5 and 8 and their

dependent claims should be deemed allowable for at least the same reasons. Further, to the extent claim 16 recites similar subject matter, it should be deemed allowable for at least the same reasons.

**Claim 2**

Claim 2 recites “wherein said order [of the data update requests] is an order in which said data update requests were made.” The Examiner asserts that paragraph 109 of Pudipeddi teaches this aspect of the claim. The aspect of Pudipeddi cited by the Examiner discloses that requests for data recall are queued sequentially and are processed in the order received. However, there is no teaching or suggestion that the data items 406 (cited for teaching the claimed data requests) are in an order in which the data requests were made. In particular, this would be contrary to the operation of Pudipeddi. Pudipeddi is directed to making queues for each medium that contains a request. By the use of queues, data recall can be achieved more efficiently since the mounting and dismounting of a medium is not required and shuttling back and forth on a same medium is not required. See paras. 49 and 50. Therefore, the purpose of Pudipeddi is to organize the requests for data. Therefore, the order of the data requests in queues 402 are not in an order in which the data requests were made.

Consequently, claim 2 should be deemed allowable. To the extent claims 6 and 9 recite similar subject matter, claims 6 and 9 should also be deemed allowable for at least the same reasons.

**Claim 3**

Claim 3 recites “said capacity corresponds to a number of said data update requests

which said data processing unit is adapted to optimally process in order in said one block before processing said data update requests in said next one of said blocks.” The Examiner asserts that paragraph 40 of Goudie teaches this aspect of the claim. The paragraph of Goudie cited by the Examiner discloses:

The `empty blocks` field is a byte wide field that indicates the number of spare (unused) blocks on the tail of a single queue, although some of the blocks may be used by an uncommitted write process which is still loaded). A single queue contains a maximum of 255 blocks, which enables the `empty blocks` and `used blocks` fields to be accessed together. The value in `empty blocks` multiplied by the `block size` gives a good approximation of the storage space (RAM) currently available, although this excludes any space available in the block which `committed tail block` points to.

There is no teaching or suggestion that a capacity corresponds to a number of data update requests which the data processing unit is adapted to optimally process in order in said one block before processing said data update requests in said next one of said blocks, as claimed.

Consequently, claim 3 should be deemed allowable. To the extent claims 7 and 10 recite similar subject matter, claims 7 and 10 should be deemed allowable for at least the same reasons.

#### **IV. Rejection of claims 11-14 under 35 U.S.C. § 103**

Claims 11-14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi in view of Goudie and in further view of In re Harza, 274 F.2d 669, 671, 124 USPQ 378, 380 (CCPA 1960). To the extent claim 11 recites subject matter similar to claim 1, it should be deemed allowable for at least the same reasons. Further, dependent claims 12-14 should also be deemed allowable for at least the same reasons.

**V. Rejection of claim 15 under 35 U.S.C. § 103**

Claim 15 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Pudipeddi. To the extent claim 15 recites subject matter similar to claim 1, it should be deemed allowable for at least the same reasons.

**VI. New Claims**

Applicant has added claims 17-19 to provide a more varied scope of protection. Claims 17-19 should be deemed allowable by virtue for their dependency to claim 1 for at least the reasons set forth above. Moreover, the art cited by the Examiner does not teach the elements of claims 17-19.

**VII. Conclusion**

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

AMENDMENT UNDER 37 C.F.R. § 1.111  
U.S. Appl. No. 10/802,471

**Attorney Docket No. Q99144**

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

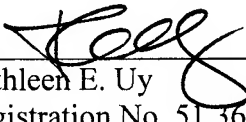
Respectfully submitted,

SUGHRUE MION, PLLC  
Telephone: (202) 293-7060  
Facsimile: (202) 293-7860

WASHINGTON OFFICE

**23373**

CUSTOMER NUMBER

  
Ruthleen E. Uy  
Registration No. 51,361

Date: January 29, 2007